REMARKS

This Amendment is in response to the Office Action mailed October 15, 2002. The Office Action objected to the Figures, rejected claims 29, 30, 39-41, and 45-48 under 35 U.S.C. § 102, and rejected claims 31, 42-44, and 49-51 under 35 U.S.C. § 103. Applicants have amended claims 29, 30, and 39. Claims 29-31 and 39-51 remain pending in the application. Reconsideration in light of the amendments and remarks made herein is respectfully requested.

Drawings

The Office Action indicated that the drawings were objected to because of the quality of the copies submitted was not adequate. Based on a telephone communication with examiner Christian La Forgia on January 7, 2003, it was determined that the drawings were objected to because the faxed drawings received by the PTO were of poor quality.

Applicants herein resubmit a clear set of drawings with this response. Applicants believe that this set of drawings are in compliance with 37 C.F.R 1.84. The Draftperson & Examiner are respectfully requested to withdraw their objection to the drawings.

Rejections Under 35 U.S.C. § 102

13. The Office Action rejected claims 29, 30, and 45 under 35 U.S.C. § 102(e) as being anticipated by Tremblay et al. (U.S. Patent No. 6,021,469) (hereinafter Tremblay '469).

The Office Action asserts that Tremblay '469 teaches every limitation claimed in independent claims 29 and 30, and dependent claim 45.

Applicants respectfully disagree.

One aspect of the present invention improves the efficiency of execution of a virtual machine by having the virtual machine refer to information about increase and/or decrease in an operand stack and/or basic blocks generated by the virtual machine compiler.

14. The rejection of claim 29 asserts that Tremblay '469 teaches the claimed limitation of "the storage method being characterized by storing each virtual machine instruction in the virtual machine instruction sequence associated with different succeeding instruction information, the succeeding instruction information for a given virtual machine instruction indicating a change in a storage state of data in a stack due to execution of a virtual machine instruction executed after the given virtual machine instruction."

However, the cited sections of Tremblay '469 (Figures 1, 2, & 4A; column 5, line 36 to column 6, line 28; column 7, lines 30-57) do not teach "storing each virtual machine instruction in the virtual machine instruction sequence associated with different succeeding instruction information, the succeeding instruction information for a given virtual machine instruction indicating a change in a storage state of data in a stack due to execution of a virtual machine instruction executed after the given virtual machine instruction" as claimed.

As illustrated in the First Embodiment (starting on page 58 of the patent application) and Figure 47 of the present application, a virtual machine of the present invention stores the value in the SOS variable 122 on the top of the operand stack 123 after the execution of the virtual machine instruction "U/Push3". The virtual machine of the present invention does not change the contents of the stack pointer SP 112 and the operand stack 123 after the execution of the virtual machine instruction "D/Push4". Thus, by identifying when an increase and/or decrease to the operand stack are to occur, the present invention can more efficiently execute virtual machine instructions.

For instance, according to one aspect of the invention implemented in a four-stage pipeline, both an increase and decrease of the operand stack can be identified in a DECODE

stage, and therefore the execution of the virtual instructions become more efficient by skipping a WRITE-BACK stage. In Tremblay '469, on the other hand, it is unknown what a succeeding instruction will be at the time a virtual instruction is executed, and therefore all stages in the four-stage pipeline must be executed, including the WRITE-BACK stage. (See Tremblay, Fig. 3).

Thus, Applicants submit that Tremblay '469 does not disclose the claimed element where succeeding instruction stages can be executed more efficiently by prior knowledge of subsequent instruction stages.

16. The rejection of claim 30 asserts that Tremblay '469 teaches the claimed limitation of "each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas."

However, for the reasons discussed above with reference to claim 29, the cited sections of Tremblay '469 (Figures 1, 2, & 4A; column 5, line 36 to column 6, line 28; column 7, lines 30-57) do not teach or suggest generating succeeding instruction information that indicates if a virtual machine instruction executed after the associated virtual machine instruction is an instruction for increasing the level of the operand stack or decreasing the level of the operand stack. The claimed elements (including "each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas") perform such task. Thus, Applicants submit that Tremblay '469 does not teach the claimed invention.

21. The rejection of claim 45 asserts that Tremblay '469 teaches all claimed limitations in independent claim 30 and the additional limitation in claim 45.

As discussed above, Tremblay '469 does not teach or suggest every claimed limitation of the parent claim 30. Thus, as a result of its dependence on independent claim 30 and for

the reasons discussed above with reference to claim 30, Applicants submit that Tremblay '469 also does not teach the limitations of dependent claim 45.

Applicants respectfully request the withdrawal of the rejection of claims 29, 30, and 45 under 35 U.S.C. § 102(e) as being anticipated by Tremblay '469.

22. The Office Action rejected claims 39-41 and 46-48 under 35 U.S.C. § 102(e) as being anticipated by Wahbe et al. (U.S. Patent No. 6,151,618) (hereinafter Wahbe '618).

The Office Action asserts that Wahbe '618 teaches every limitation claimed in claims 39-41 and 46-48.

Applicants respectfully disagree.

23. The rejection of claim 39 asserts that Wahbe '618 teaches the claimed limitation of "...stores a virtual machine instruction sequence generated by compiler ...."

Applicants submit that Wahbe '618 does not teach or suggest that the claimed tasks are performed by a compiler as claimed. As shown by step 612 in Fig. 6 of Wahbe '618, the virtual machine of Wahbe '618 divides virtual machine codes into NO-BRANCH INSTRUCTION BLOCKS (equivalent to basic blocks in the present invention). On the other hand, as illustrated by the Fifth Embodiment of the present invention (starting on page 98 of the patent application) the virtual machine codes are divided into basic blocks in advance by a virtual machine compiler. The result of the division into the basic blocks is notified to the virtual machine using the claimed method for storing virtual machine codes of the present invention. Accordingly, the virtual machine does not have to divide the virtual machine codes into the basic blocks as in Wahbe '618. This permits a virtual machine to execute various optimization processes in a shorter length of time.

Claims 40-41 and 46-48 are similarly rejected as being anticipated by Wahbe '618. As a result of their dependence on independent claim 39 and for the reasons discussed above with reference to claim 39, Applicants submit that Wahbe '618 does not teach the limitations of claims 40-41 and 46-48.

Applicants respectfully request withdrawal of the rejection of claims 39-41 and 46-48 under 35 U.S.C. § 102(e) as being anticipated by Wahbe '618.

Rejection Under 35 U.S.C. § 103

31. The Office Action rejected claim 31 under 35 U.S.C. § 103(a) as being unpatentable over Tremblay '469 in view of Rupp (U.S. Patent No. 4,177,514) (hereinafter Rupp '514).

Applicants respectfully disagree.

The rejection of claim 31 asserts that Tremblay '469 teaches the claimed invention including "wherein each virtual machine instruction in the virtual machine instruction sequence is associated with a set of succeeding instruction information that indicates a change in a storage state of the data in the stack means due to execution of a virtual machine instruction executed after the associated virtual machine instruction."

However, as discussed in detail above with reference to claim 29, Tremblay '469 does not teach or suggest that succeeding instruction stages can be executed more efficiently by prior knowledge of subsequent instruction stages as claimed. For the reasons set out above as to claim 29, Applicants submit that even in view of Rupp '514, Tremblay '469 fails to teach every limitation claimed.

Applicants respectfully request withdrawal of the rejection of claim 31 under 35 U.S.C. § 103(a) as being unpatentable over Tremblay '469 in view of Rupp '514.

39. The Office Action rejected claims 42-44 and 49-51 under 35 U.S.C. § 103(a) as being unpatentable over Wahbe '618 in view of Tremblay '469.

Applicants respectfully disagree.

While Applicants disagree that Wahbe '618 in view of Tremblay '469 teach or suggest the limitations in claims 42-44 and 49-51, this argument need not be reached. Because dependent claims 42-44 and 49-51 all stem from patentably distinguishable independent claim 39 (see above discussion of claim 39), Applicants submit that Wahbe '618 in view of Tremblay do not teach or suggest all limitations of claims 42-44 and 49-51.

Applicants respectfully request withdrawal of the rejection of claims 42-44 and 49-51 under 35 U.S.C. § 103(a) as being unpatentable over Wahbe '618 in view of Tremblay '469.

VERSION WITH MARKINGS TO SHOW CHANGES MADE

29. (Amended) A storage method used by instruction storing means that stores a virtual machine instruction sequence generated by compiler to be executed by a virtual machine, having a stack architecture, ~~under control of a real machine,~~

the storage method being characterized by storing each virtual machine instruction in the virtual machine instruction sequence associated with different succeeding instruction information, the succeeding instruction information for a given virtual machine instruction indicating a change in a storage state of data in a stack due to execution of a virtual machine instruction executed after the given virtual machine instruction.

30. (Amended) A storage method used by instruction storing means that stores a virtual machine instruction sequence generated by compiler to be executed by a virtual machine ~~under control of a real machine~~,

wherein the storage method results in:

the instruction storing means being a plurality of instruction blocks that constitute the virtual machine instruction sequence, the instruction blocks corresponding to basic blocks;

the instruction blocks each including:

an identifier area for storing an identifier that specifies a start position of the instruction block in the instruction storing means;

a non-branch instruction area for storing non-branch instructions belonging to a corresponding basic block; ~~and~~

a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block; and

each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas.

31. A computer-readable recording medium that stores a program to have a computer function as a virtual machine with a stack architecture,

wherein the virtual machine comprises:

stack means for temporarily storing data in a last-in first-out format;

instruction storing means for storing a virtual machine instruction sequence and a plurality of sets of succeeding instruction information, wherein each virtual machine instruction in the virtual machine instruction sequence is associated with a set of succeeding instruction information that indicates a change in a storage state of the data in the stack means due to execution of a virtual machine instruction executed after the associated virtual machine instruction;

read means for reading a virtual machine instruction and an associated set of succeeding instruction information from the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to a combination of the read virtual machine instruction and the read set of succeeding instruction information.

39. (Amended) A storage method used by instruction storing means that stores a virtual machine instruction sequence generated by compiler to be executed by a virtual machine ~~under control of a real machine~~,

wherein the storage method results in:

the instruction storing means being a plurality of instruction blocks that constitute the virtual machine instruction sequence, the instruction blocks corresponding to basic blocks;

the instruction blocks each including:

an identifier area for storing an identifier that specifies a start position of the instruction block in the instruction storing means;

a non-branch instruction area for storing non-branch instructions belonging to the corresponding basic block; and

a branch instruction area for storing at least one branch instructions belonging to the corresponding basic block.

Conclusion

In view of the amendments and remarks made above, it is respectfully submitted that the pending claims are in condition for allowance, and such action is respectfully solicited. Authorization is hereby given to charge our Deposit Account No. 19-2814 for any charges that may be due. Furthermore, if an extension is required, then Applicants hereby request such an extension.

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231 on January 14, 2003.

By: Sandy Malec

Sandy Malec

Signature

Dated: January 14, 2003

Very truly yours,

SNELL & WILMER L.L.P.

Joseph W. Price
Registration No. 25,124
1920 Main Street, Suite 1200
Irvine, California 92614-7230
Telephone: (949) 253-4920